



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/994,497	11/26/2001	Arvind Srinivasan	15437-0541	1924
29989	7590	01/24/2005	EXAMINER	
HICKMAN PALERMO TRUONG & BECKER, LLP				TRAN, AMY
2055 GATEWAY PLACE				ART UNIT
SUITE 550				PAPER NUMBER
SAN JOSE, CA 95110				2157

DATE MAILED: 01/24/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/994,497	SRINIVASAN ET AL.	
	Examiner	Art Unit	
	Amy Tran	2157	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 26 November 2001.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-38 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-38 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date <u>3-25-02/08-04-04</u> .	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
	6) <input type="checkbox"/> Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed on November 26, 2001.

Claims 1-38 are pending examination. Claims 1-38 represent method and system for dynamic reconfiguration of applications on a server.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-7, 10-16, 18-26, 29-34, and 36-38 are rejected under 102(e) for being unpatentable over Nock et al. (hereafter referred as Nock) US Patent No. 6,513,115.

As to claim 1, Nock teaches a computer system on which one or more applications execute, wherein a first application configuration is defined by first application configuration information, the first application configuration providing

for execution of the application, a computer-implemented method for reconfiguring the application without restarting the computer system, comprising: reading second application configuration information defining a reconfigured version of the application; constructing a second application configuration based on the second application configuration information (column 4 lines 22-33, configuration information is being updated); and providing the second application configuration to an application runtime environment for servicing new requests related to the application while maintaining the first application configuration for servicing existing requests related to the application (column 4 lines 21-63, column 5 lines 60-67, the server process continues to operate and reside in the memory, while its configuration information is being updated).

As to claim 2, Nock teaches the method of claim 1, Nock further teaches destroying the first application upon completion of all application service requests using the first application configuration, whereby computer system resources used to maintain the first application configuration are made available for other uses (column 4 lines 24-27, column 6 lines 37-46, the server process continues to service the requests from clients, while updating information involves accessing the data structure containing the current information and manipulating the information to conform to the configuration request).

As to claim 3, Nock teaches the method of claim 1, Nock further teaches assessing session information related to an existing application user session to use for servicing new requests from the same user during the existing application user session (column 6 lines 51-53, column 8 lines 30-41, reconfigured server process continues servicing users' requests without restarting the system, and user 's requests can be new requests from the same user during the existing application user session).

As to claim 4, Nock teaches the method of claim 1, Nock further teaches determining that the second application configuration successfully initialized prior to providing the second application configuration to the runtime environment for servicing new requests (column 6 lines 46-53, once the configuration information update is satisfied, a status message is generated before the reconfigured server process starts to accept a request from a client).

As to claim 5, Nock teaches the method of claim 1 and claim 4, wherein the step of determining that the second application configuration successfully initialized is based on a communication from the application runtime environment (column 6 lines 46-53, once the configuration information update is satisfied, a status message is generated before the reconfigured server process starts to accept a request from a client).

As to claim 6, Nock teaches method of claim 1, Nock further teaches receiving a request to reconfigure the computer system; and reconfiguring the computer system in response to the request (column 5 lines 60-67, upon accepting the configuration request, the new configuration information included with the request is used to update the old configuration information of the existing server process).

As to claim 7, Nock teaches the method of claim 1, wherein constructing the second application configuration is further based on an application runtime environment configuration (column 5 line 60-67, column 6 lines 35-53, existing server process continues to operate while its configuration information is updated).

As to claim 10, Nock teaches the method of claim 1, Nock further teaches logging one or more messages related to providing the second application configuration to the computer system (column 6 lines 46-53, once the configuration information update is satisfied, a status message is generated before the reconfigured server process starts to accept a request from a client).

As to claim 11, Nock teaches the method of claim 1, wherein providing the second application configuration comprises:
updating a current configuration reference to reference the second application configuration rather than the first application configuration (column 6 lines 47-53,

the reconfigured server process becomes an existing server process which starts to accept clients' requests).

As to claim 12, Nock teaches the method of claim 1 and claim 11, wherein updating the current configuration reference is performed atomically (column 5 lines 60-67, method 200 of updating the current configuration reference avoids a conflict between two server processes operating simultaneously with different configuration parameters).

As to claim 13, Nock teaches the method of claim 1, wherein the first application configuration is maintained for servicing, without interruption, existing requests related to the application from an existing connection (column 4 lines 24-27, column 5 lines 60-67, the server process continues servicing clients' requests, while its configuration information is updated).

As to claim 14, Nock teaches a computer system on which one or more applications execute, wherein a former application configuration is defined by a first application configuration information and a current application configuration is defined by a second application configuration information, the application configurations providing for execution of the application, a computer-implemented method for processing application service requests, comprising: completing processing a pending first request for an application service that is associated with the former application configuration (column 4 lines 24-27,

column 6 lines 37-46, the server process continues to service the requests from clients, while its configuration information is updated); receiving a second request for an application service; associating the second request with the current application configuration; accessing the second application configuration information defining the current application configuration; and processing the second request according to the second application configuration information (column 5 lines 60-67, column 6 lines 1-53, once the configuration information update is satisfied, the reconfigured server process becomes existing server process and starts to accept a request from a client).

As to claim 15, Nock teaches the method of claim 14, wherein the pending first request and the second request originate from the same user session, and wherein completing processing the pending first request and processing the second request are performed without restarting the computer system (column 4 lines 22-27, column 6 lines 47-53, the server process continues to operate users' requests while its configuration information is updated, the complete reconfigured server process then continues servicing users' requests without restarting the system).

As to claim 16, Nock teaches the method of claim 14, Nock further teaches determining whether the first application configuration information is referenced

by at least one pending process (column 4 lines 30-33, determining whether an existing server process is executing); and deleting the first application configuration information from the computer system in response to determining that the first application configuration information is not referenced by at least one pending process (column 4 lines 24-27, column 6 lines 37-46, the server process continues to service the requests from clients, while its configuration information updating involves accessing the data structure containing the current information and manipulating the information to conform to the configuration request).

As to claim 18, Nock teaches the method of claim 14 and claim 16, wherein determining whether the first application configuration information is referenced by at least one pending process comprises:

determining whether a reference count is equal to a predetermined value indicating that no processes are referencing the first application configuration information (column 4 lines 27-28, lines 30-32, determination whether or not existing server process is executing).

As to claim 19, Nock teaches a computer system on which one or more applications execute, wherein a plurality of application configurations are defined by a plurality of associated application configuration information and one application configuration of the plurality of application configuration is the current configuration associated with a current version of a first application and two or

more other application configurations of the plurality of application configurations are former configurations associated with former versions of the first application, a computer-implemented method for processing application service requests, comprising:

completing processing one or more pending requests for application services that are associated with any of the former configurations according to former configuration information that is associated with each of the one or more pending requests (column 4 lines 24-27, column 6 lines 37-46, the server process continues to service the requests from clients, while its configuration information is updated);

upon receiving a new request for an application service, accessing current application configuration information defining the current configuration; and processing the new request according to the current application configuration information (column 5 lines 60-67, column 6 lines 1-53, once the configuration information update is satisfied, the reconfigured server process becomes existing server process and starts to accept a request from a client).

As to claim 20, Nock teaches a computer system on which one or more applications execute, wherein a current application configuration is defined by a first application configuration information, the current configuration providing for execution of the application, a computer-readable medium comprising instructions which, when executed by one or more processors, cause the one or

more processors to reconfigure the application without restarting the computer system (fig 1), by performing the steps of:

reading second application configuration information defining a reconfigured version of the application (column 4 lines 22-33, configuration information is being updated);

constructing a second application configuration based on the second application configuration information; and providing the second application configuration to an application runtime environment for servicing new requests related to the application from a new connection while maintaining the first application configuration for servicing existing requests related to the application from an existing connection (column 4 lines 21-63, column 5 lines 60-67, the server process continues to operate and reside in the memory, while its configuration information is being updated).

As to claim 21, Nock teaches the system of claim 20, wherein execution of the instructions by the one or more processors causes the one or more processors (fig 1) to perform:

destroying the first application configuration upon completion of all application service requests using the first application configuration, whereby computer system resources used to maintain the first application configuration are made available for other uses (column 4 lines 24-27, column 6 lines 37-46, the server process continues to service the requests from clients, while updating information

is involves accessing the data structure containing the current information and manipulating the information to conform to the configuration request).

As to claim 22, Nock teaches the system of claim 20, wherein execution of the instructions by the one or more processors causes the one or more processors (fig 1) to perform:

accessing session information related to an existing application user session to use for servicing new requests from the same user during the existing application user session (column 6 lines 51-53, column 8 lines 30-41, reconfigured server process continues servicing users' requests without restarting the system, and user 's requests can be new requests from the same user during the existing application user session).

As to claim 23, Nock teaches the system of claim 20, wherein execution of the instructions by the one or more processors causes the one or more processors (fig 1) to perform:

determining that the second application configuration successfully initialized prior to providing the second application configuration to the runtime environment for servicing new requests related to the application (column 6 lines 46-53, once the configuration information update is satisfied, a status message is generated before the reconfigured server process starts to accept a request from a client).

Art Unit: 2157

As to claim 24, Nock teaches the system of claim 20 and claim 23, wherein execution of the instructions by the one or more processors causes the one or more processors (fig 1) to perform the step of determining that the second application configuration successfully initialized based on a communication from the application runtime environment (column 6 lines 46-53, once the configuration information update is satisfied, a status message is generated before the reconfigured server process starts to accept a request from a client).

As to claim 25, Nock teaches the system of claim 20, wherein execution of the instructions by the one or more processors causes the one or more processors (fig 1) to perform:

receiving a request for reconfiguring the computer system absent any changes to the computer system configuration; and reconfiguring the computer system based on the second application configuration in response to the request (column 5 lines 60-67, upon accepting the configuration request, the new configuration information included with the request is used to update the old configuration information of the existing server process).

As to claim 26, Nock teaches the system of claim 20, wherein constructing a second application configuration is based on the second application configuration information and an application runtime environment configuration, and wherein execution of the instructions by the one or more processors causes the one or more processors to perform providing the second application configuration to the

server for servicing new requests related to the application according to the second application configuration (fig 1, column 5 line 60-67, column 6 lines 35-53, the server process continues to operate and reside in the memory while its configuration information is updated, the reconfigured server process becomes an existing server process which starts to accept clients connections).

As to claim 29, Nock teaches the system of claim 20, wherein execution of the instructions by the one or more processors causes the one or more processors (fig 1) to perform:

logging one or more messages related to providing the second application configuration to the computer system (column 6 lines 46-53, once the configuration information update is satisfied, a status message is generated before the reconfigured server process starts to accept a request from a client).

As to claim 30, Nock teaches the system of claim 20, wherein execution of the instructions by the one or more processors causes the one or more processors (fig 1) to perform providing the second application configuration by performing the steps of:

updating a current configuration reference to reference the second application configuration rather than the first application configuration (column 6 lines 47-53, the reconfigured server process becomes an existing server process which starts to accept clients ' requests).

As to claim 31, Nock teaches the system of claim 20 and claim 30, wherein execution of the instructions by the one or more processors causes the one or more processors to perform updating the current configuration reference atomically (fig 1, column 5 lines 60-67, method 200 avoid a conflict between two server processes operating simultaneously with different configuration parameters).

As to claim 32, Nock teaches a computer system on which one or more applications execute, wherein a former application configuration is defined by a first application configuration information and a current application configuration is defined by a second application configuration information, the application configurations providing for execution of the application, a computer-readable medium comprising instructions which, when executed by one or more processors, cause the one or more processors (fig 1) to process application service requests, by performing the steps of: completing processing a pending first request for an application service that is associated with the former application configuration (column 4 lines 24-27, column 6 lines 37-46, the server process continues to service the requests from clients, while its configuration information is updated); receiving a second request for an application service; associating the second request with the current application configuration; accessing the second application configuration information defining the current application configuration; and processing the second request according to the second

application configuration information (column 5 lines 60-67, column 6 lines 1-53, once the configuration information update is satisfied, the reconfigured server process becomes an existing server process which starts to accept a request from a client).

As to claim 33, Nock teaches the system of claim 32, wherein the pending first request and the second request originate from the same user session, and wherein execution of the instructions by the one or more processors causes the one or more processors to perform completing processing the pending first request and processing the second request without having to restart the computer system (fig 1, column 4 lines 22-27, column 6 lines 47-53, the server process continues to operate users' requests while its configuration information is updated, the complete reconfigured server process then continues servicing users' requests without restarting the system).

As to claim 34, Nock teaches the system of claim 32, wherein execution of the

instructions by the one or more processors causes the one or more processors (fig 1) to perform the steps of:

determining whether the first application configuration information is referenced by at least one pending process (column 4 lines 30-33, determining whether an existing server process is executing); and

deleting the first application configuration information from the computer system in response to determining that the first application configuration information is

not referenced by at least one pending process (column 4 lines 24-27, column 6 lines 37-46, the server process continues to service the requests from clients, while updating information involves accessing the data structure containing the current information and manipulating the information to conform to the configuration request).

As to claim 36, Nock teaches the system of claim 32 and claim 34, wherein execution of the instructions by the one or more processors causes the one or more processors (fig 1) to perform determining whether the first application configuration information is referenced by at least one pending process by performing; determining whether a reference count associated with the first application configuration information is equal to a predetermined value indicating that no processes are referencing the first application configuration information (column 4 lines 27-28, lines 30-32, determination whether or not existing server process is executing).

As to claim 37, Nock teaches the system of claim 32, 34 and 36, wherein execution of the instructions by the one or more processors causes the one or more processors (fig 1) to perform determining that the second application configuration successfully initialized based on a communication from an application runtime environment (column 6 lines 46-53, once the configuration

information update is satisfied, a status message is generated before the reconfigured server process starts to accept a request from a client).

As to claim 38, Nock teaches an apparatus on which one or more applications execute, wherein a first application configuration is defined by first application configuration information, the first application configuration providing for execution of the application, the apparatus comprising:

a network interface (fig 1, column 3 lines 29-44); a memory (fig 1, 55-67); and one or more processors connected to the network interface and the memory (fig 1, lines 55-67), the one or more processors configured for reading second application configuration information defining a reconfigured version of the application; constructing a second application configuration based on the second application configuration information (column 4 lines 22-33, configuration information is being updated); and providing the second application configuration to an application runtime environment for servicing new requests related to the application while maintaining the first application configuration for servicing existing requests related to the application (column 4 lines 21-63, column 5 lines 60-67, the server process continues to operate and reside in the memory, while its configuration information is being updated).

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 8, 9, 27, 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nock in view of Prasad et al. (hereinafter "Prasad") US Patent 6,539,381.

As to claim 8, Nock teaches the method of claim 1 and claim 7, wherein constructing the second application configuration. Nock doesn't explicitly teach the limitation "timestamp". However, Prasad teach a system and method for synchronizing database information over communications network distributed among a plurality of servers (see abstract). Prasad teaches using a timestamp as a unique time value indicating a time at which database information is created or modified (column 13 line 58-column 14 line 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Nock by using timestamp to determine if one application configuration information different from the other application configuration information. One would be motivated to do so to check revision of application configuration information and identify which version of application configuration information is the most recent one.

As to claim 9, Nock teaches the method of claim 1, wherein constructing the second application configuration. Nock doesn't explicitly teach the limitation "timestamp". However, Prasad teaches a system and method for synchronizing database information over communications network distributed among a plurality of servers (see abstract). Prasad teaches using a timestamp as a unique time value indicating a time at which database information is created or modified (column 13 line 58-column 14 line 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Nock by using timestamp to determine if one class file has been changed. One would be motivated to do so to check revision of a class file and identify which version of a class file is the most recent one.

As to claim 27, Nock teaches the system of claim 20 and claim 26, wherein execution of the instructions by the one or more processors causes the one or more processors to perform constructing the second application configuration. Nock doesn't explicitly teach the limitation "timestamp". However, Prasad teaches a system and method for synchronizing database information over communications network distributed among a plurality of servers (see abstract). Prasad teaches using a timestamp as a unique time value indicating a time at which database information is created or modified (column 13 line 58-column 14 line 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Nock by using timestamp to determine if one application

configuration information different from the other application configuration information. One would be motivated to do so to check revision of application configuration information and identify which version of application configuration information is the most recent one.

As to claim 9, Nock teaches the method of claim 1, wherein execution of the instructions by the one or more processors causes the one or more processors to perform constructing the second application configuration. Nock doesn't explicitly teach the limitation "timestamp". However, Prasad teaches a system and method for synchronizing database information over communications network distributed among a plurality of servers (see abstract). Prasad teaches using a timestamp as a unique time value indicating a time at which database information is created or modified (column 13 line 58-column 14 line 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Nock by using timestamp to determine if one class file has been changed. One would be motivated to do so to check revision of a class file and identify which version of a class file is the most recent one.

6. Claims 17 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nock in view of Croft et al. (hereinafter Croft) US Patent 6,470,436.

As to claim 17, Nock the method of claims 1 and claim 16, wherein the first application configuration information occupies one or more blocks of memory

(column 3 lines 60-67). Nock doesn't explicitly teach releasing memory blocks. However, Croft teaches a hardware or software apparatus, or a combination of both, is used for efficiently managing the dynamic allocation, access and release of memory (see abstract). Croft teaches releasing of memory used in a computational environment (column 9 lines 40-62).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Nock by managing the dynamic allocation, access and release of unused memory. One would be motivated to do so to utilize available memory space reliably.

As to claim 35, Nock the method of claims 32 and claim 34, wherein the first application configuration information occupies one or more blocks of memory (column 3 lines 60-67), and wherein execution of the instructions by the one or more processors causes the one or more processors to perform deleting the first application configuration (column 4 lines 24-27, column 6 lines 37-46, the server process continues to service the requests from clients, while updating information involves accessing the data structure containing the current information and manipulating the information to conform to the configuration request). Nock doesn't explicitly teach releasing memory blocks. However, Croft teaches a hardware or software apparatus, or a combination of both, is used for efficiently managing the dynamic allocation, access and release

of memory (see abstract). Croft teaches releasing of memory used in a computational environment (column 9 lines 40-62).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Nock by managing the dynamic allocation, access and release of unused memory. One would be motivated to do so to utilize available memory space reliably.

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

- Gorczyca et al. US Patent 5,822,531 discloses system and method for dynamically reconfiguring a clusters of computer systems.
- Frey, Jr., Alexander H. US Patent 6,714,949 discloses dynamic file system configurations.
- Chu, Clare US Patent 6,182,119 discloses dynamically configurable filtered dispatch notification systems.
- Marshall et al. US Patent 5,784,563 discloses method and system automated reconfiguration of a client computer or user profile in a computer network.
- Brow, Alan E. US Patent 5,727,208 discloses method and apparatus for configuration of processor operating parameters.

Contact Information

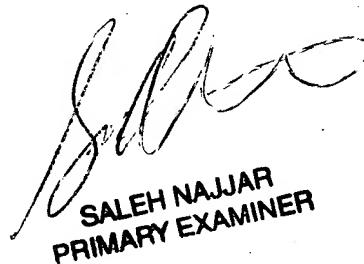
8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Amy Tran whose telephone number is (571) 272-4243. The examiner can normally be reached on M-F from 8:30am to 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571) 272-4001. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

at

January 19, 2005



SALEH NAJJAR
PRIMARY EXAMINER